di.me: Ontologies for a Pervasive Information System

Simon Scerri[1], Ismael Rivera[1], Jeremy Debattista[1], Simon Thiel[2], Keith Cortis[1], Judie Attard[1], Christian Knecht[2], Andreas Schuller[2], and Fabian Hermann[2]

[1]INSIGHT Centre for Data Analytics, Galway, Ireland
firstname.lastname@deri.org
[2]Fraunhofer-Gesellschaft zur Forderung der angewandten Forschung e.V., Stuttgart, Germany
firstname.lastname@iao.fraunhofer.de

Abstract. The di.me userware is a pervasive personal information management system that successfully adopted ontologies to provide various intelligent features. Supported by a suitable user interface, di.me provides ontology-driven support for the i) integration of personal information from multiple personal sources, ii) privacy-aware sharing of personal data, iii) context-awareness and personal situation recognition, and iv) creation of personalised rules that operate over live events to provide notifications, effect system changes or share data.

1 Introduction

Di.me is an example of a pervasive information system as described in [5], i.e., a dynamic environment composed of multiple personal information sources, that is capable of perceiving contextual information and supporting mobility. The di.me functionality, as included in the freely-available open-source system[1], demonstrates the value of using ontologies as representation formats for a wide variety of abstract concepts and information elements deriving from multiple devices and online accounts.

The above claim is validated particularly by the personalisable Rule Manager, which builds atop the other three ontology-driven features to enable people to construct *email filter*-style rules that extend to their entire digital sphere. If the current user interface (UI) is extended, these rules could enable users to design rules that switch a device to 'Silent' mode when entering the office/cinema, change their online presence to 'Available' when leaving, or provide notifications when someone refers to them on the social network post. Before providing examples of supported rules, and the other di.me features, we first provide a short overview of the ontologies that drive them.

2 Modelling the Personal Information Sphere

Many vocabularies in the di.me Ontology Framework were initially engineered for the Social Semantic Desktop[2]. In the digital.me[3] (di.me) project, they were extended in

[1] http://vmuscs05.deri.ie:8443/dime-communications/static/ui/dime/index.html
[2] http://nepomuk.semanticdesktop.org/
[3] http://dime-project.eu/

three directions to cover additional: a) personal sources (beside one desktop), b) personal information (e.g., microposts and social network interactions, placemarks, etc.) and c) domains (e.g., privacy, trust, context and presence, histories and rules). Below is an overview[4] of the main vocabularies used to represent the personal di.me knowledge base, and an indication of how they enable the described di.me features.

Information gathered from personal sources is semantically lifted, i.e., transformed into an ontology-based representation of the Personal Information Model, as an instance of the PIMO Ontology. This process is in part supported by metadata crawlers at the integration stage of personal devices or online accounts, and in part by listeners which detect item creation/modification/deletion items in the entire personal information sphere. Thus, the semantic model maintains an integrated and up-to-date personal knowledge base, based on data stored on devices and online accounts, represented by the Device (DDO) and Account (DAO) ontologies. The extracted information items are represented by the Information Element (NIE) domain ontologies, which model files (NFO), events (NCAL), addressbooks (NCO), messages (NMO), etc., on personal devices; as well as social network data and activities (DLPO) [6]. The Contact ontology (NCO) is also the model behind the profile integration described in Section 3. The Sharing ontology (NSO) extends the Privacy Preference Ontology[5] to represent the concepts needed for the functionality described in Section 4, supported by the Annotation Ontology's (NAO) modelling of agent trust and item privacy levels.

The Context ontology (DCON) attaches context-dependant semantics to various items, e.g. a specific file is being modified, a specific person is nearby. To enable the situation recognition feature described in Section 5, a single DCON instance maintains a centralised representation of a person's activities, e.g., events (NCAL instances) extracted from a calendar service, files being edited and applications running (NFO instances), locations checked-in and people tagged on a social network (DLPO instances), etc. The Presence ontology (DPO) facilitates the interpretation of context information having a broad range (e.g., temperature, date/time) by mapping discrete values retrieved from sensors to pre-defined categories (e.g., 'Hot', 'Late Evening', 'Weekend'). To compare context information at different times, the user History ontology (DUHO) persists time-stamped DCON instances in separate, non-conflicting named graphs.

Finally, the context-driven rule functionality described in Section 6 is enabled by the Rule Management Ontology (DRMO). Through the UI, a large part of the above mentioned Personal Information Model elements can be wrapped within DRMO instances as filters. When di.me detects that all filters apply, it triggers the desired user-defined actions. The UI allows for great flexibility, and both concepts (e.g., person) and instances (e.g., a specific known person) can be selected as rule filters. Filters can be custom-described depending on their type, e.g., when selecting a specific person (a PIMO instance) as a filter, one can specify that the person must be nearby (a DCON property). However, if the person concept is selected (signifying 'any person'), it can be further constrained, e.g., to only apply to members of a specific group (a PIMO property).

---

[4] For in-depth ontology descriptions please refer to: http://www.semanticdesktop.org/ontologies/. For a documentation of the ontologies' integration in di.me, please refer to: http://github.com/dime-project/meta/wiki/Ontology-Framework

[5] http://vocab.deri.ie/ppo#

3 Multi-source Personal Information Integration

An example of ontology-driven information integration in di.me is provided by the Person/Contact merging feature, which in contrast to other methods, does not rely solely on syntactic similaries. It also takes into account the semantics attached to the various profile attributes derived from multiple sources (e.g. address book, social networks) [3]. When linking additional sources to di.me, e.g., a social network account, profiles for all contacts are semantically lifted onto the NCO ontology, and then compared to existing profiles. Matches that are identified are suggested for merging, as shown in Fig. 1-left.

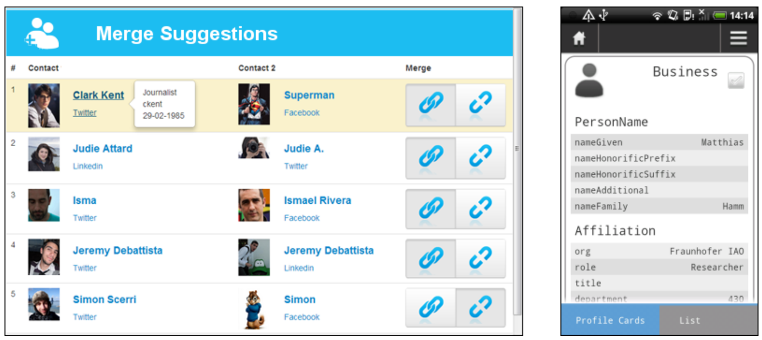

Fig. 1. Duplicate Contact Detection (Web UI) and a 'Business' Profile Card (mobile UI)

4 Privacy-aware Sharing of Personal Data

The NSO ontology enables the representation of two virtual concepts: *Databoxes* and *Profile Cards*. Databoxes are able to represent ad-hoc, user-created collections of PIM items (e.g., documents, images, archive files, etc.), organised by context (e.g., related to a topic, project, or task). Similarly, Profile Cards refer to subsets of the user's personal identification attributes (e.g., names, pictures, email addresses, etc.), manually designed for different purposes. Both Databoxes and Profile Cards can be shared through a system of whitelists and blacklists (e.g., share with a group of people, but exclude some members). An example of a custom profile card, or alternate personal identity, targeted towards business contacts is shown in Fig. 1-right.

To counteract risks associated with the increased cognitive complexity of sharing modularised data in this manner, di.me provides various warnings based on person trust levels and item privacy levels (NAO Ontology). Both levels can be adjusted manually through the UI, but also automatically based on perceived sharing activities and social network interactions (trust changed) and on provenance (privacy level changed).

5 Situation Training and Recognition

Personal situations can be saved on the spot and labelled accordingly (e.g. "Working@Office"). On saving, registered context elements are automatically added to the situation (DCON instance), each with an initial weight of zero. These weights are automatically adjusted whenever a suggested situation is confirmed by the user [2]. Future versions of di.me could allow the removal of elements deemed irrelevant to the situation (e.g., Sunny weather), and also manually modify their weights.

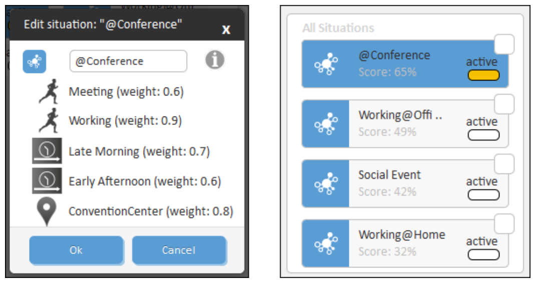

Fig. 2. Situation Management and Suggestions

Fig. 2 shows the UI components responsible for visualising a specific situation (left), and the situation suggestion bar (right). As indicated by the varied weights, the '@Conference' situation has already been trained, and the elements which have the relevance are the detected 'Working' activity (0.9) and the 'Convention Centre' placemark (0.8). The situation suggestion bar shows ranked results (by % score) of the context matching algorithm [1], which continously compares the live context information with the stored situations. Through this bar, users can manually confirm which situations are recurring. Situations are only automatically activated when matching scores are greater than 80%.

6 Personalisable Context-driven Rules

The di.me Rule Manager is accessible through the Settings tab. It is a 'lego-like' UI that enables owners to drag-and-drop objects and apply filters. Fig. 3 shows the steps required to build the following rule: *IF* (friends are nearby) & (I'm at a social event) & (I take a photo) *THEN* [Ask if I want to share it with my friends].

Step one shows objects that may be used as rule blocks. In this example, the 'Nearby People' is dragged to the corresponding coloured-box on the left-hand side. The Rule Manager then loads relevant filters. In this case (Step 2), 'Nearby Group' is selected. Available groups are then shown, and a specific one is selected (Step 3). This constitutes the first completed block. Step 4 shows how the rule looks like after two other blocks are added: i) the situation social event is active, and ii) a new image (photo) is created. To finalise the rule, an action is selected (Step 5). An accompanying message is required for notification purposes when the rule is triggered.

Each saved rule is stored as an instance of DRMO. At runtime, di.me builds a rule network for efficient rule matching. The system continously listens out for each context and system event, and checks these against the network. Rules are trigerred when the event processor detects that an entire path in the network has been activated [4]. Notification for the triggered rule is shown in Fig. 4.

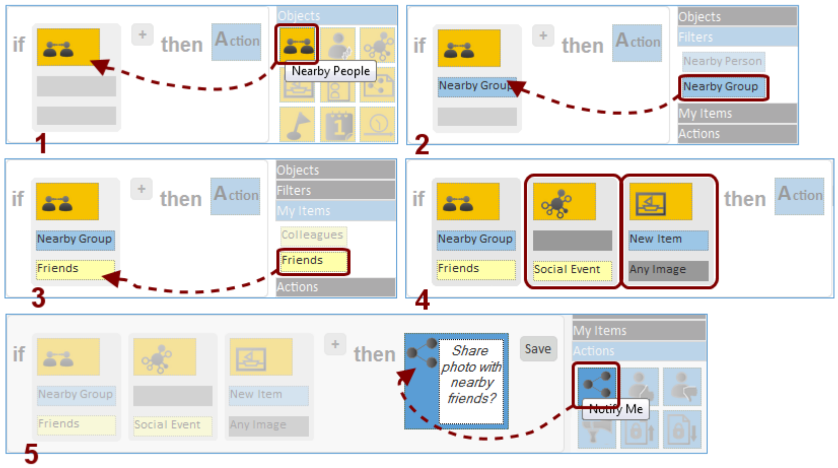

Fig. 3. The steps required to create a custom rule through the lego-like Rule Manager

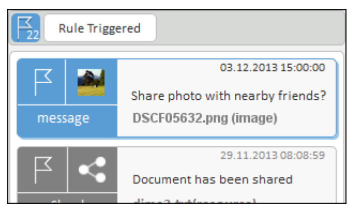

Fig. 4. Notification for a Triggered Rule

Acknowledgments

Supported by the European Commission under FP7/2007-2013 (*digital.me* ICT-257787).

References

1. J. Attard, S. Scerri, I. Rivera, and S. Handschuh. Ontology-based situation recognition for context-aware systems. In *Proceedings of the 9th international I-Semantics conference*, 2013.
2. J. Attard, S. Scerri, I. Rivera, and S. Handschuh. User preference learning for context-aware situation identification. In *Proceedings of the 5th International Workshop on Acquisition, Representation and Reasoning with Contextualized Knowledge (ARCOE-LogIC)*, 2013.
3. K. Cortis, S. Scerri, I. Rivera, and S. Handschuh. An ontology-based technique for online profile resolution. In *Social Informatics*, volume 8238 of *Lecture Notes in Computer Science*, pages 284–298. Springer International Publishing, 2013.
4. J. Debattista, S. Scerri, I. Rivera, and S. Handschuh. Processing ubiquitous personal event streams to provide user-controlled support. In *Proceedings of the 14th International Conference on Web Information System (WISE)*, 2013.
5. P. E. Kourouthanassis and G. M. Giaglis. A design theory for pervasive information systems. In *IWUC*, pages 62–70, 2006.
6. S. Scerri, K. Cortis, I. Rivera, and S. Handschuh. Knowledge discovery in distributed social web sharing activities. In *Making Sense of Microposts (#MSM2012)*, pages 26–33, 2012.